COS 513: Monte Carlo Markov chain sampling (I)

Lawrence W. Manning

29 November 2010

Introduction

We have seen several latent variable models so far (Factor analysis, Hidden Markov models, mixture models), as well as Tree Propagation, an algorithm for *exact* inference. However, there are a number of interesting cases in which exact inference methods are computationally infeasible. Today we discuss methods for *approximate* inference.

1 Approximate Inference

Example 1: Let $\mu \sim N(0, \lambda),\ x_n \sim N(\mu, \sigma^2)\ \ \forall\, n = 1, \ldots, N$, as shown in Figure 1 below. If $p(\mu)$ is Gaussian, then $p(\mu \mid x_{1:N})$ is also Gaussian.

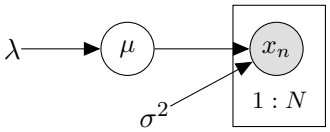

Figure 1: Graphical Model Representation of Example 1

Example 2: Now suppose that $x_{1:N}$ came from a K-mixture. (See Figure 2 for the graphical model representation.)

$$\begin{aligned} \mu_k &\sim N(0, \lambda) \quad \forall k = 1, \ldots, K \\ z_n &\sim Mult(\pi) \quad \forall n = 1, \ldots, N \\ x_n \mid z_n &\sim N(\mu_{z_n}, \sigma^2) \end{aligned}$$

Here, $p(\mu_1, \ldots, \mu_K \mid x_{1:N})$ is <u>not</u> easy to compute. Suppose that π is fixed and $K = 3$:

$$p(\mu_1, \mu_2, \mu_3 \mid x_{1:N}) = \frac{p(\mu_1, \mu_2, \mu_3, x_{1:N})}{p(x_{1:N})}, \tag{1}$$

where:

$$p(\mu_1, \mu_2, \mu_3, x_{1:N}) = p(\mu_1)p(\mu_2)p(\mu_3) \prod_{n=1}^{N} \left(\sum_{k=1}^{3} \pi_k p(x_n \mid \mu_k) \right) \tag{2}$$

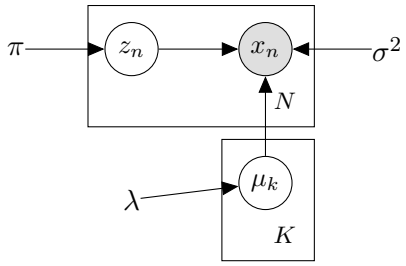

Figure 2: Graphical Model Representation of Example 2

and where:

$$p(x_{1:N}) = \int_{\mu_1}\int_{\mu_2}\int_{\mu_3} p(\mu_1)p(\mu_2)p(\mu_3)\prod_{n=1}^{N}\sum_{i=1}^{3}\pi_i p(x_n \mid \mu_i).$$

Or equivalently, marginalizing out the z_n's instead of the μ_k's:

$$p(x_{1:N}) = \sum_{z_{1:N}}\left(\prod_{k=1}^{K}\left(\int_{\mu_k} p(\mu_k)\prod_{n:z_n=1} p(x_n \mid \mu_{z_n})\right)\right) \tag{3}$$

Notice, however, that the quantity in Equation 3 is difficult to compute, since the outside sum is over every possible configuration of $z_{1:N}$. Thus, exact inference in this case is $\mathcal{O}(K^N)$.

Practical Bayesian models usually require approximate posterior inference. *Sampling methods* are one kind of approximation. Recall Equation 2:

$$\begin{aligned} p(\mu_1,\mu_2,\mu_3,x_{1:N}) &= p(\mu_1)p(\mu_2)p(\mu_3)\prod_{n=1}^{N}\left(\sum_{k=1}^{3}\pi_k p(x_n \mid \mu_k)\right) \\ &= \sum_{z_n} p(x_n, z_n \mid \mu_{z_n}) \\ &= \sum_{z_n} p(z_n)p(x_n \mid z_n, \mu_{z_n}) \\ &= \sum_{z_n} \pi_{z_n} p(x_n \mid \mu_{z_n}) \end{aligned}$$

If the parameters are unobserved, then clusters assignments are interdependent. That is, assuming that a given point is in a certain cluster changes the likelihood that other points are in that cluster.

2 Sampling methods

Denote the target distribution by $p(x)$. ($p(x)$ could be a posterior distribution or anything else.) **Idea** – approximate $p(x)$ with a set of samples:

$$p(x) \approx \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x),$$

where $x^{(i)}$ are samples from the target.

2.1 Rejection sampling

Suppose that we can compute p but cannot sample from it. However, there exist some q and M such that we can sample from q and:

$$p(x) \leq Mq(x) \; \forall x.$$

Algorithm 1 Rejection sampling

for $i = 1, \ldots, N$ **do**
  sample $x^{(i)} \sim q(x)$
  sample $u \sim Unif(0, 1)$
  if $u < \frac{p(x^{(i)})}{Mq(x^{(i)})}$ **then**
    Accept sample $x^{(i)}$ and increment i

Rejection sampling has the clear limitation that one must be able to compute $p(x)$.

2.2 Importance sampling

Suppose (still) that we can compute $p(x)$. Suppose (also) that $q(x)$ has the same support as $p(x)$ but no other restrictions. Then for arbitrary f:

$$\begin{aligned}
\mathbb{E}[f(x)] &\triangleq \int_x f(x)p(x)dx \\
&= \int_x f(x)p(x)\frac{q(x)}{q(x)}dx \\
&\approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})w(x^{(i)}), \qquad (4)
\end{aligned}$$

where $x^{(i)} \sim q(x)$ and

$$w(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}.$$

In Equation 4, q and w are referred to as the "proposal distribution" and "importance weight", respectively.

3 Markov Chain Monte Carlo (MCMC)

MCMC lets us collect samples from a wide class of distributions, scales well with dimension, and we only need to know $p(x)$ up to some constant Z. That is:

$$p(x) = \frac{\tilde{p}(x)}{Z}$$

In Bayesian applications, this condition is quite often satisfied (e.g. Equation 1).

Idea – draw a sample x^* from the proposal distribution q, and accept according to some criterion, which may be random.

Algorithm 2 Metropolis algorithm (1953)

Require: $q(x_1 \mid x_2) = q(x_2 \mid x_1)$
  loop
    sample $x^{(t+1)} \sim q$, given $x^{(t)}$
    sample $u \sim Unif(0, 1)$
    $A(x^{(t+1)}, x^{(t)}) \leftarrow \min\left(1, \frac{p(x^{(t+1)})}{p(x^{(t)})}\right)$
    if $u < A(x^{(t+1)}, x^{(t)})$ **then**
      Accept $x^{(t+1)}$
    else
      $x^{(t+1)} \leftarrow x^{(t)}$

If $q(x_1 \mid x_2) > 0$ for all $x_1,\ x_2$, then

$$p_m(x^{(t)}) \rightarrow p(x) \quad \text{as} \quad t \rightarrow \infty$$

General idea behind MCMC – define a Markov chain on x whose *stationary* distribution is the target distribution.

Recipe –

1. Run Markov chain for a long time (forever, if possible)
2. Collect samples at some lag (forever, if possible)